PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04K 1/00, H04L 9/00 — A1

(11) International Publication Number: **WO 00/11828**

(43) International Publication Date: 2 March 2000 (02.03.00)

(21) International Application Number: PCT/US99/19061

(22) International Filing Date: 18 August 1999 (18.08.99)

(30) Priority Data:

60/096,935	18 August 1998 (18.08.98)	US
not furnished	17 August 1999 (17.08.99)	US

(71) Applicant *(for all designated States except US)*: RLJ TIME-STAMP CORPORATION [US/US]; 10439 Lone Oak Lane, Los Altos Hills, CA 94020 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: BILDOS, Ahto [EE/EE]; Akadeemia tee 21, EE–12618 Tallinn (EE). LAUD, Peeter [EE/EE]; Sopruse pst 211–15, EE–Tallinn (EE). LIPMAA, Helger [EE/EE]; Akadeemia tee 10–21, EE–12618 Tallinn (EE). VILLEMSON, Jan [EE/EE]; Sopruse 8–49, EE–Tallinn (EE).

(74) Agents: SHERIDAN, James, A. et al.; Flehr Hohbach Test Albritton & Herbert LLP, Suite 3400, 4 Embarcadero Center, San Francisco, CA 94111–4187 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.

(54) Title: TIME–STAMPING WITH BINARY LINKING SCHEMES

(57) Abstract

A digital signature certification system creates a nonce and attaches a time to the nonce to create a time stamped nonce uniquely identifying the time stamp then attaches the time stamped nonce to a document, attaches a digital signature to the document, then attaches a time to the document to form a time stamped document, so that the nonce uniquely identifies the signature on the document.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

TIME-STAMPING WITH BINARY LINKING SCHEMES

[BLL V98] Ahto Buldas, Peeter Laud, Helger Lipmaa, Jan Villemson, "Time-stamping with binary linking schemes," Proc. CRYPTO '98.

[BdM91] Josh Benaloh, Michael de Mare, "Efficient broadcast time-stamping," Technical report 1, Clarcson University Department of Mathematics and Computer Science, August 1991.

[BHS92] Dave Bayer, Stuart Haber, W. Scott Stornetta, "Improving the efficiency and reliablity of digital time-stamping," In Sequences '91: Methods in Communications, Security, and Computer Science, pp. 329-334. Springer-Verlag, 1992.

[HS91] Stuart Haber, W. Scott Stornetta, "How to time-stamp a digital document," Journal of Cryptology, 3 (2):99-111, 1991.

[HS97] Stuart Haber, W. Scott Stornetta, "Secure names for bit-strings," In proceedings of the 4th ACM Conference on Computer and Communications Security, pp. 28-35, April 1997.

Field of the Invention

This invention relates to digital signatures in computer documents, and more particularly to time stamping digital signatures so that the latest time will be unambiguously known.

Background

Time stamping is a set of techniques enabling the ascertaining of when an electronic document was created or signed. The real importance of time-stamping comes about with the legal use of long lifetime documents. A problem with time stamping signed documents comes about when, for example, the signer repudiates the document and the cryptographic primitives become unreliable. The security of

the signature becomes questionable. For example, a signer might claim she had lost her signature key, repudiate the signing, and bring the authenticity of a signature into question in order to escape responsibility for a document.

Recently, especially in the local regulation of digital signatures, organizational and legal questions about reliability in time stamping signatures have been gaining world wide attention. In the prior art, in addition to defining the responsibilities of the owner of the signature, the duties and responsibilities of the Time Stamping Service (TSS) employed must be stated. It is becoming increasingly important that trust of the TSS not be an issue; or that questions relating to the need to trust the TSS be minimized. In order to make users liable only for their own actions, the offender in a situation involving a digital signature infraction must be positively identifiable, even if the offender is the TSS.

Digital signatures, since they are administered by systems that inherently do not have any relation to physical time (real time) in their operation, do not have real time acknowledgments. For this reason, the association of an electronic document directly to a unique moment in time is difficult, and may be impossible. The best we can do with time stamping is Relative Temporal Authentication (RTA), that is, we can associate a document with some relative time that we trust.

This method, which is often used, is based on a complexity-theoretic assumption of the existence of collision-resistant one-way hash functions. RTA gives the verifier with two time stamped documents the ability to verify which of the two was created first.

The following examples of existing time stamping systems will illustrate the problems:

1) An example of an existing time stamping technique is a simple time stamping protocol. The TSS appends the current time t to the current document X, the composite document is signed, and two values, t and $s=sig_{TSS}(t,X)$ are returned to the client. A weakness of this approach is the unreliability of documents with old time stamps after a signature key leakage, which may make it impossible to verify the time t on the document. This implies that for a reasonable solution the TSS must be unconditionally trusted. It is therefore widely accepted that a secure time stamping system cannot rely solely on the keys or on any other secret information of that sort.

2) One example of an embodiment of a digital signature certification system of the type discussed above is shown in [BHS92,HS97] and Patent No. 5,136,646 by Haber and Stornetta. Signatures with time certificates attached are linked together in a one-way function, such that the verifier is able to follow a step by step chain of intermediate time stamps, and is able to ascertain at each step which was created earlier. In this way a type of time tree is grown, with the credibility of the signature verified by trusted documents preceding and following in time.

The time certificate for the n-th submitted document is:

$c=D_{TSS}(n,t_n,ID_n,X_n,L_n)$, where t_n is the current time, ID_n is the identifier of the submitter, and L_n is the n-th catenate certificate defined by the recursive formula:

$Ln=(t_{n-1},ID_{n-1},X_{n-1},H(L_{n-1}))$, and H is a collision-resistant one-way hash function.

There are several complications with the implementation of the above system. The number of steps needed to verify the one-way relationship between two time stamps is linear with respect to the number of time stamps between them, so a single verification may be as costly as creating an entire chain.

It was pointed out in the publication of the Benolah-de Mare proposal [BdM91] that this solution has impossible trust and broadcast requirements. A modification was proposed [HS91] wherein, every time stamp is linked with k>1 time stamps directly preceding. This variation decreases the requirements for broadcasting but increases the space required for storing individual time stamps.

3) Tree linking systems as disclosed [in BdM91, BHS92, HS97] US Patent Number Re. 34,954 reduce verification cost in a significant way.

[BHS92] illustrated in Fig. A]. The time-stamping procedure is divided into rounds. The time-stamp R_r for round r is a cumulative hash of the time stamp R_{r-1} for round r-1 and of all the documents submitted to the TSS during the round r. After the end of the r-th round a binary tree T_r is built. Every participant P_i who wants to time-stamp at least one document in this round, submits to the TSS a hash $y_{i,i}$ which is a hash of all the documents he wants to time-stamp in this round. The leaves of T_r are labeled by the submitted data items y_i. Each inner node k of T_r is recursively labeled by numerical values $H_k := H(H_{kL}, H_{kR})$, where k_L and k_R are correspondingly the left and the right child nodes of k, and H is a collision-resistant hash function. The TSS has to store only the time-stamps R_r for rounds (Fig. 1). All the remaining information, required to verify whether a certain document was time-stamped during a fixed round is included into the time certificates.

A time certificate of a document comprises the information required to verify whether a certain document was time stamped during a fixed round, i. e., for restoring the label of the predecessor node needed to know the labels of the sibling nodes. For example, the time certificates for y_3 in Figure 1 is $(r;(y_4,L),(H_4,R))$. The verifying procedure of the time stamp of y_3 consists of verifying the equality:

$$R_r = H(H(H_4, H(y_3, y_4)), R_{r-1}).$$

The size of the time certificate and thereby also the number of computational steps during the verification is logarithmic on the number of documents submitted. The values of R_r are stored into a database and some of them are published in a newspaper.

The schemes are feasible but provide the RTA for the documents issued during the same round only if we unconditionally trust the TSS to maintain the order of time-stamps in T_r. Therefore, this method either increases the need for trust or otherwise limits the maximum temporal duration of rounds to the insignificant units of time (one second in Digital Notary system). However, if the number of submitted documents during a round is too small, the expenses of time-stamping a single document may become unreasonably large.

Summary of the Invention

The present invention comprises a method of time-stamping a digital document using a binary linking scheme where the value of the catenate certificate L_n is generated by applying a one-way hash function H to a catenation comprising the value of the catenate certificate L_{n-1} and the value of another suitably chosen catenate certificate $L_{f(n)}$, with f being a fixed deterministic function algorithm, i.e.

$$L_n = H(n, X_n L_{n-1}, L_{f(n)}).$$

With choosing the function f appropriately it is possible to verify a one-way relationship between two time-certificates with a number of computational steps proportional to the logarithm of the number of time-stamped documents. A function f is presented that guarantees logarithmic verification.

A binary linking scheme is presented where the linking function f is chosen in such a way that it satisfies the anti-monotonic property, i.e. that $f(m)<n<m$ implies $f(n) \geq f(m)$. Said property is sufficient for the existence of a series $n(1),...,n(k),...$ of indices such that, for each k, the time-certificate $L_{n(k)}$ is generated only using the values of L_j, where $n(k-1)<i<n(k)$, and of $L_{n(j)}$ with $j<k$. Thus, the intervals between the issuance of different $L_{n(k)}$ can be thought about as the rounds. The anti-monotonic property says that the time-stamp for a round is not linked directly to the inner time-stamps of other rounds.

A method is also presented of certifying the moment of signing, not only the moment of submitting. Before signing a document X a principal P generates nonce N and time-stamps it. By a nonce is meant sufficiently long random bit-string, such that the probability it has been already time-stamped is negligible. Principle P then includes the time-stamp $L(N)$ of N to the document, signs it and obtains the time-stamp $L(S)$ of the signature $S=D_p(L(N),X)$. For the verification of the document X, the verifier has to compare both these time-stamps with the time-stamps trusted by the verifier (which may be nonces generated by the verifier herself). As there are one-way dependencies between $L(N)$, S and $L(S)$ the verifier may conclude that the signature was created in the time-frame between the moments of issuance of $L(N)$ and of $L(S)$ respectively. If these moments are close enough, the signing time can be ascertained with necessary precision. In this solution there are no supplementary duties to the TSS or to the other principals.

A time-stamping procedure is also defined, as follows: (1) the client sends to the TSS the data item X to be time-stamped; (2) the TSS answers immediately by sending then current L_n and the necessary data for verifying the one-way dependency between L_n and the time-stamp for the previous round. The TSS signs L_n and sends the signature $D_{TSS}(n, L_n)$ to the client; (3) if the round is over, the client may apply the TSS for the data necessary to verify a one-way relationship between L_n and the time-stamp for round. Therefore, the TSS is not able to rearrange the time-stamps

during a round. This means the present scheme reduces the need for trusting the TSS in maintaining the temporal order of time-stamped documents.

Brief Description of the Drawings

Fig. 1 is flow chart of a tree linking system for the certification of Digital Signatures.

Fig. 2 is flow chart of a binary linking system (BLS) for the certification of Digital Signatures.

Fig. 3 is flow chart of a BLS with the shortest verification links between digital signatures.

Fig. 4 is a flow chart of an Accumulated Linking System (ALS) which may be used in the invention.

Fig. 5 is flow chart of a Time Stamp system of the invention.

Table I is a definition of a recursive linking system for digital signature verification.

Table II shows how recursive linking may be programmed on a computer.

Table III is a proof that a further reduction in the complexity of linking digital signatures is not feasible beyond the invention.

Table IV-A and IV-B comprise proofs of the sufficiency of the invention for verification of digital signatures as disclosed.

Description of the Preferred Embodiment

In the following a definition is given of time-stamping systems applicable in legal situations. Later the approach will be justified and compared to older systems.

A time-stamping system consists of a set of principals with the time-stamping server (TSS) together with a triple (S, V, A) of protocols. The stamping protocol S allows each participant to post a message. The verification protocol V is used by a principal having two time-stamps to verify the temporal order between those time-stamps. The audit protocol A is used by a principal to verify whether the TSS carries out his duties. Additionally, no principal (in particular, TSS) should be able to produce fake time-stamps without being caught.

A time-stamping system has to be able to handle time-stamps which are anonymous and do not reveal any information about the content of the stamped data. The TSS is not required to identify the initiators of time-stamping requests.

The present notion of a time-stamping system differs from the one given in, e.g., [BdM91] in several important aspects. The differences are explained below.

Relative Temporal Authentication:

The main security objective of time-stamping is temporal authentication - ability to prove that a certain document has been created at a certain moment of time. Although the creation of a digital data item is an observable event in the physical world, the moment of its creation cannot be ascertained by observing the data itself. The best one can do is to check the relative temporal order of the created data items (i.e., prove the RTA) using one-way dependencies defining the arrow of time, analogous to the way in which the growth of entropy defines the arrow of time in the physical world. For example, if H is a collision-resistant one-way hash function, one can reliably use the following "rough" derivation rule: if H(X) and X

are known to a principal P at a moment t, then someone (possibly P itself) used X to compute H(X) at a moment prior to t. Preferably, the system utilizes collision-resistant one-way hash functions.

Definition 1. A collision-resistant one-way hash function is a function H which has the properties of compression, ease of computation, preimage resistance, 2nd-preimage resistance and collision resistance.

Definition 2. Let p be a binary relation on N, such that x p y implies $x < y$ and H to be a collision-resistant one-way hash function. A (p, H)-linking scheme is a procedure to link a family (H_n) of data items together using auxiliary linking items L_n satisfying the recursive formula

$$L_n := H(H_n, L_n\,1, \ldots, Ln_{\xi p\text{-}1(n)}),$$

where $n1 \geq \ldots \geq n_{\xi p\text{-}1(n)}$ are exactly the elements of $\rho^{-1}(n) := (m \mid m\ \rho\ n)$ (the preimage of n by ρ). A sequence $(m_i)^{\xi}_{i=1}$, where $m_i\ \rho\ m_{i+1}$ is called a verifying chain between m_1 and m_ξ with length ξ.

In the context of time-stamping $H_n = H(n,X_n)$, where X_n denotes the n-th time-stamped document. The linking item L_n is also referred to as a time-stamp of X_n. Note that a one-way relationship between L_n and L_m $(n < m)$ does not prove that in the moment of creating X_n the bit-string X_m did not exist, but we do know that X_n did exist at the moment of creating L_m.

We have omitted the t_n in the formula for H_n, whereas it should not be taken for granted that the value t_n indeed represents the submission time of X_n. The only way for a principal to associate a time-stamp with a certain moment of time is to time-stamp a nonce at this moment. By a nonce we mean a sufficiently long random bit-string, such that the probability it has been already time-stamped is negligible. In

order to verify the absolute creating time of a document time-stamped by another principal, the verifier has to compare the time-stamp with the time stamps of nonces generated by the verifier herself. In this solution there are neither supplementary duties to the TSS nor to the principals. The use of nonces illustrates the similarity between time-stamping and ordinary authentication protocols, where nonces are used to prevent the possible reuse of old messages from previous communications.

By using RTA it is possible to determine not only the submitting time of the signature but also the time of signing the document. Before signing a document X the principal P generates a nonce N and time-stamps it. He then includes the time-stamp L(N) of N to the document, signs it and obtains the time-stamp L(σ) of the signature $\sigma=sig_p(L(N), X)$. From the view-point of the TSS these stamping events are identical (he need not be aware whether he is time-stamping a nonce or meaningful data). For the verification of the document X, the verifier has to compare both these time-stamps with the time-stamps trusted by her. As there are one-way dependencies between L(N), σ and L(σ) the verifier may conclude that the signature was created in the time-frame between the moments of issuance of L(N) and of L(σ) respectively. If these moments are close enough, the signing time can be ascertained with necessary precision.

3.2 Detection of Forgeries

A time-stamping system must have properties enabling users to verify whether an arbitrary time-stamp is correct or not. Possession of two documents with corresponding time-stamps is not enough to prove the RTA between the documents because everyone is able to produce fake chains of time-stamps.

A time-stamping system should allow the user (1) to determine whether the time-stamps possessed by an individual have been tampered with; and (2) in the case of tampering, to determine whether the time-stamps were tampered with by the TSS or tampered after the issuing (generally by unknown means). In the second case,

there is no one to bring an action against. The principals interested in legal use of time-stamps should themselves verify their correctness immediately after the issuing (using signatures and other techniques discussed later) because if the signature of the TSS becomes unreliable, the signed time-stamps cannot be used as evidence. In order to increase the trustworthiness of the time-stamping services it should be possible for the clients to periodically inspect the TSS. Also, in the case when the TSS is not guilty he should have a mechanism to prove his innocence, i.e., that he has not issued a certain time-stamp during a certain round.

Additionally, the TSS must publish regularly, in an authenticated manner, the time-stamps for rounds [BdM91] in mass media. If the time-stamping protocol includes (by using collision-resistant one-way hash functions) (1) the message digest of any time-stamp issued during the r-th round, into the time-stamp for r-th round, and (2) the message digest of the time-stamp for round r - 1 into any time-stamp issued during the r-th round, it will be difficult for anyone to forge a time-stamp without detection. The forgery detection procedures should be simple. Forgeries should be determinable either during the stamping protocol (when the time-stamp, signed by the TSS, fails to be correct) or later when it is unable to establish the temporal order between two otherwise correct time-stamps.

3.3 Feasibility Requirements

The time-stamping systems of [BdM91] and [HS97] use nonlinear partial ordering of time-stamps and therefore do not support RTA. A later discussion shows how to modify the linear linking scheme [HS91] to fulfill the security objectives (RTA and detection of forgeries). On the other hand, in practice, in this scheme the detection of forgeries would take too many steps. It is easy to forge time-stamps assuming that the verifier has limited computational power. This leads to the question of feasibility. In order to make RTA feasible in the case when time-stamps belong to different rounds, it is reasonable to define an additional layer of links between the time-stamps for rounds.

Definition 3. Assume (p,H) and (δ,H) linking schemes and a monotonically increasing function ξ: N→N. By a (p,ξ, δ, H)-linking scheme is meant to be a procedure for linking a family (H_n) of data items together using auxiliary linking items L_n and $\mathcal{L}_r$ satisfying the recursive formulas shown in Table I.

The values $\mathcal{L}_r$ are also referred to as the time-stamps for rounds. Note that the time-stamps requested from the TSS during the verification protocol should belong to the set of time-stamps for rounds because only these time-stamps are available in the time-stamping server.

Definition 4. A (ρ,ξ,δ,H)-linking scheme is said to be an Accumulated Linking Scheme (ALS) with rank m, if

1. If $\xi(r) < n \le \xi(r+1)$ then $p^{-1}(n) \subset [\xi(r+1), \xi(r+1)] \cup \xi(N)$.
2. $\xi(r+1) - \xi(r) \ge m$.

A (p, H)-linking scheme enables accumulated time-stamping if for arbitrary positive m there exists ξ, such that the (p, ξ, p, H)-scheme is an ALS with rank m.

If the linking scheme used enables accumulated time-stamping, the duration of the rounds can be flexibly enlarged in order to guarantee that only a negligible fraction of the time-stamps are kept in the memory of the time-stamping server.

Let n be the total number of time-stamps issued till the moment of the current run of stamping/verification protocol. The feasibility requirements can be summarized with the following:

1. The number of the evaluations of the hash function during the verification protocol should be O(log n). In particular, the number of time-stamps examined during a single run of the verification protocol should be O(log n);

2. There should be a conveniently small upper bound to the length of rounds, whereas the clients want to get their time-stamps in reasonable time. It

seems to be sensible to require that the stamping protocol of the n-th document must terminate before the TSS has received additional O(log n) time-stamp requests. In real applications it is desirable for the average length of rounds to be constant (this would guarantee that for an arbitrary constant c there would be a negligible fraction of rounds with length greater then c).

3. The size of an individual time-stamp should be small.

There is a trade-off between these quantities. Later there is presented an improvement of the scheme above.

First Version of The System: Linear Linking

For pedagogical reasons, the protocols and the basic organizational principles of the system using the linear linking scheme are outlined below. This scheme fulfills all the trust requirements but is impractical. Further, the described scheme is significantly improved by replacing the linear scheme with a binary linking scheme.

Let the number M of time-stamps per round be a constant known to the participants (clients) and all the data items X_n be of fixed size. Therefore, in the case of the linear linking scheme, the time-stamp for the r-th round has a number $\xi_r = M \cdot r$.

Role of the TSS:

The TSS maintains the following three databases:

1. the database Dc of the time-stamps of the current round.
2. the database Dp of the time-stamps of the previous round.
3. the database Dr of the time-stamps for rounds.

These databases are considered to be on-line in the sense that any client can make requests into them at any moment. The fourth database (the complete data-base of time-stamps) is also stored but not on-line (it may be stored into an archive

of CDs). Requests to this database are possible, but costly (e.g., requiring human interaction). After the end of each round, the time-stamps in D_p are stored to a separate CD (this procedure may be audited). Thereafter Dp is emptied. The time-stamp Rr for the current round is computed, added to Dr and published in a newspaper or similar publication (two processes which should be audited). The database Dc is copied into Dp and a new database Dc is created.

Stamping Protocol:

Suppose, the current round number is r.

1. Client sends X_n to the TSS.
2. The TSS finds $H_n = H(_n, X_n)$ and $L_n = (H_n, L_{n-1})$, and adds the pair $(H_n L_n)$ to Dc.
3. The TSS signs the pair (n, L_n) and sends $(n, L_n, Sig_{TSS}(n, L_n))$ back to the client.
4. The TSS sends the tuple head(n) = $(H_{n-1}, H_{n-2}, \ldots\ldots, H_{\xi r+1} + 1)$ to the client.
5. The client verifies the signature of TSS and checks whether

$$H(H_n H(H_{a-1}, \ldots\ldots, H(H_{\xi r-1} + 1, L_{\xi r-1}) \ldots\ldots)) = L_n$$

where the true values $L_{\xi r}$ can be found either from the newspaper or by requesting for their values from the on-line database D_r of the TSS.

After the M requests have been answered the TSS finishes the round by finding $L_{\xi r} = H(H'_{\xi r}, L_{\xi r-1})$(where $H'_{\xi r} = (H_{\xi r}, L_{\xi r-1})$) and publishing L_{er} and his public key K_{TSS} in the newspaper or the like. The client may now continue, during a limited period, the protocol in order to get the complete individual time-stamp for X_n.

6. The client sends a request to the TSS.
7. Let tail (n) = $(H_{\xi r-1}, H_{\xi r-2}, \ldots, H_{n+2}, H_{n+1})$. The TSS answers by sending (tail (n), sig_{TSS} (tail (n))) to the client.
8. The client checks whether

$$L_{\xi r} = H(H_{\xi r-1}, H(H_{\xi r-2}, \ldots\ldots H(H_{n+2}, H(H_{a+1}, L_n))\ldots))$$

Definition 5. The complete individual time-stamp s_n for the n-th document is

s_n:=(tail(n),head(n),n,L_n,$sig_{TSS}(n,L_n)$).

Every client who is interested in the legal use of a time-stamp, should validate it during the stamping protocol. In a relatively short period between the 1st and the 3rd step and between the 4th and 6th step, the signature key of TSS is trusted to authenticate him and therefore, his signature on an invalid head (n) or tail (n) can be used as an evidence in the court. But the client is responsible for doing it when the signature key of TSS can still be trusted. Later, the signature of TSS may become unreliable and therefore only the one-way properties can be used.

Verification Protocol:

Let r (n) denote the round where s_n was issued. Assume, the verifier has two time-stamped documents (X_m,s_m) and (X_n,s_n) where m < n.

1. The verifier checks the validity of the equations (2) and (3) for both time-stamps.

2. If r (m) = r (n) then the data held in tail (m) and head (n) will be enough to check whether

$L_n = H(H_{n3} H(H_{n-1}, ...H(H_{m+1}, L_m).....))$.

3. If r (m) < r (n), the verifier sends a request to the TSS.

4. The TSS answers by sending the tuple

$V_{mn} = (H^1_{\xi r}(n) - 1, H^1_{\xi r}(n) - 2, ..., H^1_{\xi r}(m))$

and the signature $sig_{TSS}(V_{mn})$ to the verifier.

5. The verifier validates the signature, finds $L_{\xi r(m)}$ using (3), finds L_r (n) -1 using the formula

$L_{r(n)-1} = H(H^1_{\xi r(n)-1}, H(H^1_{\xi r(m)}, L_{\xi r(m)})...))$.

and finally, compares the value of L_n in s_n with the value given by (2).

Audit Protocol:

Because of the possible legal importance of the time-stamps issued by the TSS, there should be some mechanism to audit the TSS. One easy way to do it is to

periodically ask for time-stamps from the TSS and verify them. If these time-stamps are linked inconsistently (i.e., Eq. (2) and (3) hold for both time-stamps but the verification protocol fails), the TSS can be proven to be guilty. Also, there has to be a mechanism for the TSS to prove that he has not issued a certain time-stamp S in a certain round r. This can be done if the TSS presents all the time-stamps issued using the r-th round, and the time-stamp, found by using these time-stamps and the linking rules, coincides with the published time-stamp.

Above an outline is presented of a time-stamping system that fulfills trust requirements. Next is shown how to make this system feasible by using a BLS as shown in Fig. 4.

In order to issue the individual time-stamp for the n-th document, the TSS has to find the shortest verifying chains between $\xi_{r(n)-1}$ and n and between N and $\xi_{r(n)}$. The n-th individual time-stamp consists of the minimal amount of data necessary to verify the mutual one-way dependencies between all Lj which lay on these chains. It can be shown that if f satisfies the implication

$$m > n \Rightarrow (f(m) \leq f(n) \vee f(m) \geq n)$$

then (f,H) enables accumulated time-stamping (the proof has been omitted because of its technicality.) In particular, the binary linking scheme described in enables accumulated time-stamping. For a fixed m let $k := [\log_2 m]$, $\xi_0 := 0$, $\xi_1 := 2^k - 1$ (the source of T_k) and for arbitrary $i > 1$,

$$\xi(i) := \begin{cases} \xi_{2^j} + \xi_{i-2^j}, & \text{if } i \neq 2^j \\ 2\text{-}\xi_{i/2} + 1, & \text{if } i = 2^j, \end{cases}$$

where $j := [\log_2 i]$. The length of the n-th time-stamp in this scheme does not exceed $2 \cdot 3 \cdot \log(n) \cdot x$ bits, where x is the output size of the hash function H.

The maximum length of rounds grows proportionally to O(log n). However, the average length of rounds is constant and therefore it is practical to publish the time-stamps for rounds after constant units of time. This can be achieved easily with the following procedure. If the "deadline" for a round is approaching and there are still q time-stamps not issued yet, assign random values to the remaining data items H_n.

Remark 1. Denote by ord n the greatest power of 2 dividing n. In the ALS presented above, it is reasonable to label time-stamps in the lexicographical order with pairs (n, p), where $0 \le p \le$ ord n and $n > 0$. Then,

$$f(n,p) := \begin{cases} (0,p) & n=2^p \\ (n-2^p, \text{ord}(n-2^p)), & \text{otherwise} \end{cases}$$

and $g(n, p) := (n,p-1)$ if $p > 0$ and $g(n, 0) := (n - 1, \text{ord } (n-1))$. Also, the formulas of ξ_i will simplify. In this case, $\xi(i) := (2^{k-1} i, k - 1 + \text{ord } i)$, for $i \ge 1$.

It is easy to show that for each n and m the shortest verifying chain between n and m is uniquely defined. The data υ_{mn} necessary to verify the one-way dependence is computed by the procedure TSData(m, n) as shown in Table II and illustrated in Fig. 5.

Let (f, H) be a BLS satisfying the implication (4). Let $x < y < z < w$ and C_1, C_2 be verifying chains from z to x and w to y respectively. It is obvious that C_1 and C_2 have a common element. Thus, if $m < n$ then the verifying chains tail (m) and head (n) have a common element c which implies the existence of a verifying chain.

$$(m = n_o, n_1, \ldots\ldots, n_{i-1}, n_i = C, n_{i+1},\ldots,n_{i-1},n_c = n)$$

This chain can be found by a simple algorithm and is of logarithmic length. Let (m) denote the round into which m belongs. The proof of the last claim for the case r (m) = r (n) is given below under the heading proof of Theorem 1. If m and n belong to different rounds, the verifying is straightforward, because of the similar structure of the second layer of links. the verifying chain from n to m is of the form

$$(m, ...,m',\xi_{r(m)},n',...,n).$$

where the number of ξ_j^{-B} is logarithmic due to the fact that the time-stamps for rounds are linked together in a way similar to the linking of all time-stamps (Fig. 2). The length of the sequences (m,....m') and (n', ..., n) is also logarithmic.

Example 2. For the chains given in Example 1, the common element is 7 and the verifying chain between 4 and 10 is (4, 5, 6, 7, 10).

Corollary 1. Due to the similarity between the verification and the stamping procedure, for an arbitrary pair of time-stamped documents the number of steps executed (and therefore, also the number of time-stamps examined) during a single run of the verification protocol is O(log n).

Optimality:

Our solution meets asymptotically the feasibility requirements, but could these requirements be refined? Mostly not, an insight into this is given below. Namely, we show that for any linking scheme there does not exist a time-stamping solution where (1) the length of the time-stamps is O (log n), (2) for any m and n there exists a verifying chain between m and n with the length O (log n) that is completely contained in the union $S(m) \cup S(n)$ of the corresponding individual time-stamps and (3) the stamping protocol will end in a logarithmic time.

We prove this under the assumptions (1) that an individual time-stamp is a subset of N and (2) that the size of a time-stamp is proportional to the size of $\|S(n)\| + \|p^{-1}(S(n)\| = O(\|p^{-1}(S(n)\|)$ (holds if the transitive closure p^n of p coincides with the natural order <, i.e., the time stamp S(n) consists of tail (n) and head (n))).

Theorem 2. Let p be a binary relation on N satisfying $P^n = <$. There does not exist a function S: $\|N \rightarrow 2^{IN}$ such that

1. $|\rho^{-1}(S(n))| < c_1 \log n$ for some c_1, for any n; also see Table IV-A and IV-B.

2. For every m and n there exists a p-chain($m=m_1, m_2, \ldots, m_k=n$) where $m_i = S(m) \cup S(n)$ (that is, the number of stamps to examine during the verification protocol is greater than 2).

3. For any n, $\max(S(n)) - n \leq c_2 \log n$ for some constant c_2 as shown in Table III.

The Theorem 2 can be straightforwardly generalized to claim that the number of examined time-stamps must be greater than any fixed constant.

Proof of Theorem 1:

We will prove an upper bound for the length of the verifying chain for the linking scheme described elsewhere. Let $e_k = 2^k - 1$, i.e. e_k is the number of the last vertex of T_k. To simplify the proof we add the vertex 0 to the scheme and link it with all the vertices that have less than two outgoing links. These are exactly the vertices e_K. Let L(a, b) denote the length of the shortest path between a and b. The equations L (O, e_k) = 1, L (e_{k-1} , ek) = 2 and $e_{k-1} = e_{k-1} + 1$ follow immediately from the definition.

Binary Linking Scheme:

In the current section we give a construction of a practical linking scheme with logarithmic upper bound to the length of the shortest verifying chain between any two time-stamps.

Definition 6. Let f and g be functions from N to N satisfying the condition $f(n) \leq g(n) < n$ for any n. A(f,g,h) binary linking scheme (BLS) is a (p,H) linking scheme where for any n, $p^{-1}(n)$ [=(f(n), g(n)). In order to guarantee the existence of a verifying chain between arbitrary x and y, we have to take g(n):=n-1. In these cases we omit n-1 and talk about an (f,H)-BLS.

A binary linking scheme can alternatively be defined as a directed countable graph which is connected, contains no cycles and where all the vertices have two outgoing edges (links). Let us construct an infinite family of such graphs Tk in the following way:

1. T1 consists of a single vertex which is labeled with the number 1. This vertex is both the source and the sink of the graph T1

2. Let Tk be already constructed. Its sink is labeled by 2^k-1. The graph Tk+1 consists of two copies of Tk, where the sink of the second copy is linked to the source of the first copy, and an additional vertex labeled by 2^{k+1}-1 which is linked to the source of the second copy. Labels of the second copy are increased by 2^k-1. The sink of Tk+1 is equal to the sink of t the first copy, the source of Tk+1 is equal to the vertex labeled by 2^{k+1}-1.

Thereafter, link all the vertices of the second copy which have less than two outgoing links to the source of the first copy. Note that there is now a double link from the sink of the second copy to the source of the first copy as shown in Fig. 3.

The sequence (Tk) defines a binary linking scheme, add links from the sources of any such initial segment to a special vertex labeled by 0 (Fig. 2). Here (see also Rem. 1), $f(n)=n-2^{h(n)}+1$, where h(n) is given recursively by the equation below and as illustrated in Fig. 4.

$$h(n)=\begin{cases} k, & \text{if } n=2^k-1, \\ h(n+1-2^{k-1}) & \text{if } 2^{k-1}\le n<2^k-1. \end{cases}$$

Theorem 1. Let l(a,b) be the length of the shortest verifying chain from b to a. If $k>2$ and $0<a\le b<2^k$ then $l(a,b)\le 3k-5$.

Theoretical and practical considerations of the present invention are:

1) the importance of trust of the TSS in time stamping is significantly reduced, and

2) time complexity of Relative Temporal Authentication (RTA) becomes logarithmic with the number of issued time stamps.

An embodiment of the present invention comprises a method of time stamping a digital document using binary linking. A catenate certificate L_n is generated by applying a one-way hash function H to a concatenation of the value of the catenate certificate L_{n-1} and the value of a suitably chosen catenate certificate $L_{f(n)}$, where f is a fixed deterministic function, such as:

$$L_n=H(n, X_n, L_{n-1}, L_{f(n)}).$$

The time t_n has been omitted. It should not be taken for granted that the value t_n actually represents the submission time of document X_n. With choosing the function f appropriately it is possible to verify a one-way relationship between two time certificates with a number of computational steps proportional to the logarithm

of the number of time stamped documents that are to be reviewed. A function f of the invention, which was presented at [BLLV98], guarantees logarithmic computational steps in a signature verification.

In an embodiment of the binary linking system of the invention, a linking function f, which satisfies an anti-monotonic property such as $f(m)<n<m$, which implies $f(n)>f(m)$ or $f(n)=f(m)$, is sufficient for the existence of a series $n(1),...,n(k)$. The indices are such that for each k the time certificate $L_{n(k)}$ is generated exclusively with values of L_j, where $n(k-1)<j<n(k)$, and of $L_{n(j)}$ with $j<k$. Treating intervals between the issuance of different $L_{n(k)}$ as "rounds", the anti-monotonic property insures that the time stamp for a round is not linked directly to the inner time stamps of other rounds.

In another embodiment of the invention, the moment of signing, not just the moment of submitting, is certified. Before signing a document X a principal P generates nonce N and time stamps it. A nonce is a long random bit string, with an arbitrary length judged sufficient to reduce the probability of a conflict with another time stamp to insignificance. The time stamp L(N) of N is then included in the document, the document signed, and a time stamp certification L(S) of the signature $S=D_P(L(N),X)$ results. From the standpoint of the TSS, the time stamping events are identical; that is, the TSS does not know or need to know whether the time stamping is for a nonce or for meaningful data. For the verification of the document X, the verifier compares both time stamps with other time stamps trusted by the verifier; which may be nonces developed for this purpose.

Since the dependencies between L(N), S, and L(S) are one-way, the verifier can conclude that the signature was created in the time frame between the moments of issuance of L(N) and of L(S), respectively. If these moments are close enough in time, the signing time can be ascertained with precision. In this embodiment there are no supplementary duties for the TSS or other principals.

In yet another embodiment, limited reliance on the TSS allows for a simplified system:

1) the client sends a data item X to the TSS to be time stamped,

2) the TSS responds immediately with the current L_n and the necessary data for verifying the one-way dependency between L_n and the time stamp for the previous round, signs to create an L_n, and sends the signature $D_{TSS}(n,L_n)$ to the client, and

3) if the round is over, the client may apply to the TSS for the data necessary to verify a one-way relationship between Ln and the time stamp for the round.

The above embodiment thereby reduces the need for trusting the TSS in maintaining the temporal order of time stamped documents by preventing the TSS from having an opportunity to rearrange the documents.

It will be seen that by providing time stamp verification which is independent, or at least, relatively independent, of the TSS or third parties, the integrity of the signature is significantly improved.

Definition *Assume we are given (ρ, H) and (δ, H) linking schemes and a monotonically increasing function $\xi: \mathbb{N} \to \mathbb{N}$. By a (ρ, ξ, δ, H)-linking scheme we mean a procedure for linking a family (H_n) of data items together using auxiliary linking items L_n and $\mathcal{L}_r$ satisfying the recursive formulas*

$$L_n := H(H_n, L_{n_1}, \ldots, L_{n_{\rho^{-1}(n)}}) \qquad \text{if } n \notin \xi(\mathbb{N})$$
$$\mathcal{L}_r := L_{\xi(r)} = H(\mathcal{H}_r, \mathcal{L}_{r_1}, \ldots, \mathcal{L}_{r_{\delta^{-1}(r)}})$$
$$\mathcal{H}_r := H(H_m, L_{m_1}, \ldots, L_{m_{\rho^{-1}(m)}}) ,$$

where $m = \xi(r)$, $\rho^{-1}(n) = \{m_1, \ldots, m_{\rho^{-1}(n)}\}$ $(m_1 \geq \ldots \geq m_{\rho^{-1}(n)})$ and $\delta^{-1}(r) = \{r_1, \ldots, r_{\delta^{-1}(r)}\}$ $(r_1 \geq \ldots \geq r_{\delta^{-1}(r)})$.

TABLE I

```
proc TSData(m,n) ≡
  Data := nil
  while n > m do
        Data := append(Data, H_n)
        if f(n) ≠ n − 1 ∧ f(n) ≥ m
          then Data := append(Data, L_{n−1}); n := f(n)
          else Data := append(Data, L_{f(n)}); n := n − 1
        fi
  od.
```

Here, $\text{head}(n) := TSData(\xi_{r(n-1)}, n)$ and $\text{tail}(n) := TSData(n, \xi_{r(n)})$.

Example 1. Let $\xi_0 = 0$ and $\xi_1 = 15$ (Fig. 2). In order to compute the fourth and the tenth time-stamps we need

$$\text{tail}(10) := (H_{10}, L_9, H_{14}, L_7, H_{13}, L_{12}) ,$$
$$\text{head}(10) := (H_{10}, L_9, H_7, L_6) ,$$
$$\text{tail}(4) := (H_{15}, L_9, H_{14}, L_{12}, H_7, L_9, H_6, L_3, H_5, L_4) ,$$
$$\text{head}(4) := (H_4, L_3, H_3, L_2) .$$

TABLE II

Proof. Assume that there exists such S. Let n be a sufficiently large positive integer. For a $m \in \mathbb{N}$ let $\Phi(m) := [m, m + \lceil c_2 \log m \rceil]$. The intervals $\Phi(1 + ic_2 \log n)$, $i \in 0, \ldots, \lfloor \frac{n - c_2 \log n - 2}{c_2 \log n} \rfloor$ do not intersect.

Let $m < n - c_2 \log n - 1$. In this case $\lceil m + c_2 \log m \rceil < n$. As the set $S(m) \cup S(n)$ contains a ρ-chain from m to n there should exist such $m_1 \in \Phi(m)$ and $n_1 \in S(n)$ on this chain that $m_1 \rho n_1$. Thus, for every $m < n - c_2 \log n - 1$ the set $\Phi(m) \cap \rho^{-1}(S(n)))$ is nonempty. Hence, the set $\rho^{-1}(S(n))$ has at least $\lfloor \frac{n - c_2 \log n - 1}{c_2 \log n} \rfloor = \Theta(n / \log n)$ elements. A contradiction with Condition 1. □

TABLE III

We will prove an upper bound for the length of the verifying chain for the linking scheme described in Sect. 5. Let $e_k = 2^k - 1$, i.e. e_k is the number of the last vertex of T_k. To simplify the proof we add the vertex 0 to the scheme and link it with all the vertices that have less than two outgoing links. These are exactly the vertices e_k. Let $\ell(a, b)$ denote the length of the shortest path between a and b. The equations $\ell(0, e_k) = 1$, $\ell(e_{k-1}, e_k) = 2$ and $e_k - e_{k-1} = e_{k-1} + 1$ follow immediately from the definition.

Lemma 1. *If $0 < a \le e_k \le b$ then $\ell(a, b) = \ell(a, e_k) + \ell(e_k, b)$. If $e_{k-1} \le a < e_k$ then $\ell(a, e_k) = \ell(a, e_k - 1) + \ell(e_k - 1, e_k)$.*

The claims above follow immediately from the structural properties of the linking scheme.

Lemma 2. *If $e_{k-1} \le a \le b < e_k$ then $\ell(a, b) = \ell(a - e_{k-1}, b - e_{k-1})$.*

Proof. This follows from the construction of T_k from the two copies of T_{k-1}. Here a and b are vertices in the second copy of T_{k-1} (or the last vertex of the first copy), and $a - e_{k-1}$ and $b - e_{k-1}$ are the same vertices in the first copy of T_{k-1} (or the vertex 0). □

Lemma 3. *If $0 \le a < e_k$ then $\ell(0, a) \le k$.*

Proof. Induction on k.
Base: $k = 1$. Then $a = 0$ and $\ell(0, a) = 0 < k$.
Step: $k > 1$. Observe the following cases:

- If $0 \le a < e_{k-1}$ then the induction assumption gives $\ell(0, a) \le k - 1 < k$.
- If $e_{k-1} \le a < e_k$ then $\ell(0, a) = \ell(0, e_{k-1}) + \ell(e_{k-1}, a) = 1 + \ell(0, a - e_{k-1})$ by Lemma 2. Observe the following cases:
  - $a = e_k - 1$. Then $\ell(0, a) = 1 + \ell(0, a - e_{k-1}) = 1 + \ell(0, e_{k-1}) = 2 \le k$.
  - $a < e_k - 1$. Then $\ell(0, a) = 1 + \ell(0, a - e_{k-1}) \le 1 + (k - 1) = k$ by induction assumption.

□

TABLE IV-A

Lemma 4. *If $0 < a \leq c_k$ then $\ell(a, c_k) \leq 2(k-1)$.*

Proof. Induction on k.
Base: $k = 1$. Then $a = 1$ and $\ell(a, c_k) = \ell(1, 1) = 0 = 2(k-1)$.
Step: $k > 1$. Observe the following cases:

- If $0 < a \leq c_{k-1}$ then $\ell(a, c_k) = \ell(a, c_{k-1}) + \ell(c_{k-1}, c_k) \leq 2(k-2) + 2 = 2(k-1)$ by induction assumption.
- If $c_{k-1} < a \leq c_k$ then observe the following cases:
  - $a = c_k$. Then $\ell(a, c_k) = 0 \leq 2(k-1)$.
  - $a < c_k$. Then $\ell(a, c_k) = \ell(a, c_k - 1) + \ell(c_k - 1, c_k) = \ell(a - c_{k-1}, c_{k-1}) + 1$ by the Lemma 2. Induction assumption now gives $\ell(a, c_k) = \ell(a - c_{k-1}, c_{k-1}) + 1 \leq 2(k-2) + 1 < 2(k-1)$.

□

Proof (Theorem 1). Induction on k.
Base: $k = 3$. In this case one can directly verify that $\ell(a, b) \leq 4$.
Step: $k > 3$. Observe the following cases:

- If $0 < a \leq b \leq c_{k-1}$ then the induction assumption gives us $\ell(a, b) \leq 3(k-1) - 5 < 3k - 5$.
- If $0 < a \leq c_{k-1} < b \leq c_k$ then $\ell(a, b) = \ell(a, c_{k-1}) + \ell(c_{k-1}, b) \leq 2(k-2) + \ell(c_{k-1}, b)$ by the Lemma 4. The following cases are possible:
  - $b = c_k$. Then $\ell(c_{k-1}, b) = 2 < k - 1$.
  - $b = c_k - 1$. Then $\ell(c_{k-1}, b) = 1 < k - 1$.
  - $b < c_k - 1$. Then the lemmas 2 and 3 give $\ell(c_{k-1}, b) = \ell(0, b - c_{k-1}) \leq k - 1$.

  Thus $\ell(a, b) \leq 2(k-2) + k - 1 = 3k - 5$.
- If $c_{k-1} < a \leq b \leq c_k$ then observe the following cases:
  - $b = c_k$. Then $\ell(a, b) = \ell(a, c_k) \leq 2(k-1) < 3k - 5$ by Lemma 4.
  - $b < c_k$. Then $\ell(a, b) = \ell(a - c_{k-1}, b - c_{k-1}) \leq 3(k-1) + 5 < 3k - 5$ by Lemma 2 and induction assumption.

□

As $\lceil \log b \rceil = k$ iff $c_{k-1} + 1 < b \leq c_k + 1$ we get $k < \lceil \log b \rceil + 1$ and thus

$$\ell(a, b) \leq 3\lceil \log b \rceil - 2 \ .$$

TABLE IV-B

Claims

1. A digital signature certification system comprising:
creating a nonce;
time stamping said nonce to create a time stamped nonce uniquely identifying said time stamp;
attaching said time stamped nonce to a document;
attaching a digital signature to said document with said nonce;
time stamping said document and the signature; whereby
uniquely represents said signature on said document.

2. The system of claim 1 wherein said nonce is a random bit string having a length such that the probability of an identical nonce is insignificant.

3. The system of claim 2 wherein reliance on a Time Stamping Service (TSS) for verification of a signature is reduced or eliminated.

4. The system of claim 1 wherein reliance on RTA directly with other signatures is reduced or eliminated.

5. The system of Claim 1 wherein said nonce is used as a time-related standard for RTA.

6. A digital signature certification system comprising:
creating a nonce;
time stamping said nonce to create a time stamped nonce uniquely identifying said time stamp;
attaching said time stamped nonce to a document;
attaching a digital signature to said document;
time stamping said document and the signature; whereby

the nonce stamp uniquely represents said signature on said document;
creating said time stamped nonce entries as a binary database;
linking said binary database to a verifiable RTA source; whereby
said RTA source is a verifiable point for all said time stamped nonce entries within a time frame associated with said RTA.

7. The system of claim 6 wherein said nonce is a random bit string having a length such that the probability of an identical nonce is insignificant.

8. The system of claim 7 wherein reliance on a Time Stamping Service (TSS) for verification of a signature is reduced or eliminated.

9. The system of claim 6 wherein reliance on RTA directly with other signatures is reduced or eliminated.

10. A digital signature certification system comprising:
creating a nonce means;
relating said nonce means to some time standard uniquely identifying said nonce;
attaching said nonce means to a document;
attaching a digital signature to said document and to said nonce means;
relating said document to said nonce means; whereby
said nonce means uniquely identifies said signature on said document;

11. The system of claim 10, comprising:
creating said nonce means as a database means;
linking said database means to a verifiable time; whereby
said verifiable time thereby verifying signatures associated with said nonce means within a time frame associated with said verifiable time.

12. The system of claim 10 wherein said nonce means is a data means having characteristics such that the probability of an identical nonce means is insignificant.

13. The system of claim 10 wherein reliance on commercial verification services for verification of a signature is reduced or eliminated.

14. The system of claim 11 wherein reliance on time services for verification of signatures is reduced or eliminated.

15. A method of time-stamping a digital document using a binary linking scheme where the value of the catenate certificate L_n is generated by applying a one-way hash function H to a catenation comprising the value of the catenate certificate L_{n-1} and the value of another suitably chosen catenate certificate $L_{f(n)}$, with f being a fixed deterministic function algorithm,

$$L_n = H(n, X_n, L_{n-1}, L_{f(n)}).$$

16. A method as claimed in claim 15 including verifying a one-way relationship between two time-certificates with a number of computational steps proportional to the logarithm of the number of time-stamped documents.

17. A method of digital time-stamping wherein:

each document X is given a time-certificate t(X) of reasonable length that uniquely defines the relative position of X inside the protocol-round it is time-stamped, and thereafter.

given two documents X and Y and certificates t(X) and t(Y) a verifier is able to establish a one-way relationship between the corresponding time stamps.

18. A time-stamping procedure using a binary linking scheme, comprising:

a client sends to a TSS a data item X to be time-stamped;

the TSS answers immediately by sending then current L_n and necessary data for verifying a one-way dependency between L_n and a time-stamp,

the TSS further signs L_n and sends a signed receipt $D_{\{TSS\}}$ (n, L_n) to the client and, upon completion of a round,

the client obtains the time-certificates.

19. A method of determining a time of signing a document comprising:

generating a nonce N and time-stamping the document with time-stamp L(N),

signing the document,

generating the time-stamp L(σ) of the signature $\sigma = seg_p$ (L(N), X), and verifying the document by comparing time of issuance of L(N) and L(σ).

20. A method as claimed in claim 19 wherein the time-stamp L(N) and L(σ) includes collision-resistant one way hash functions to prevent forgery of any of said time-stamps.

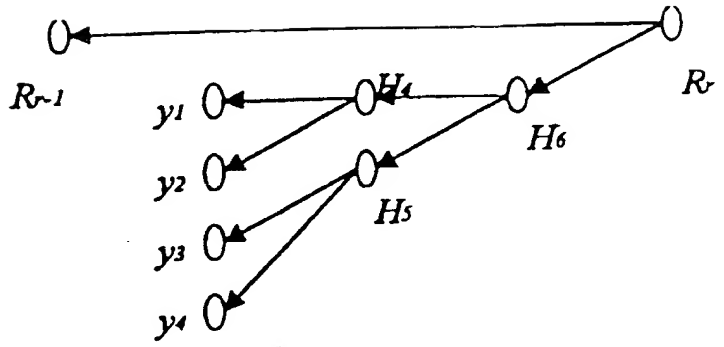

Fig.1.

PRIOR ART

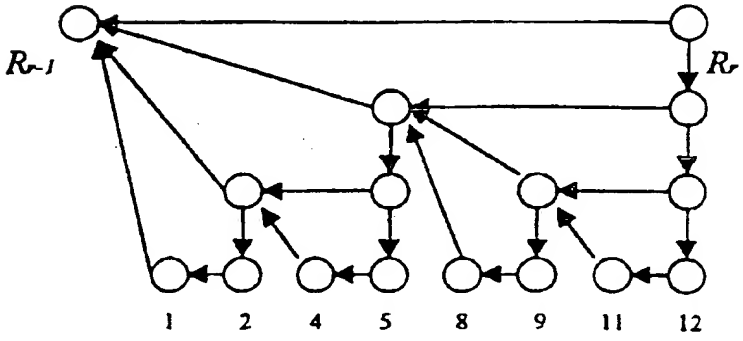

Fig. 2.

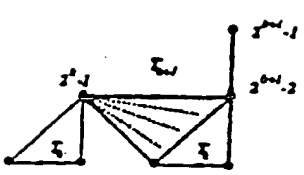

FIG 3

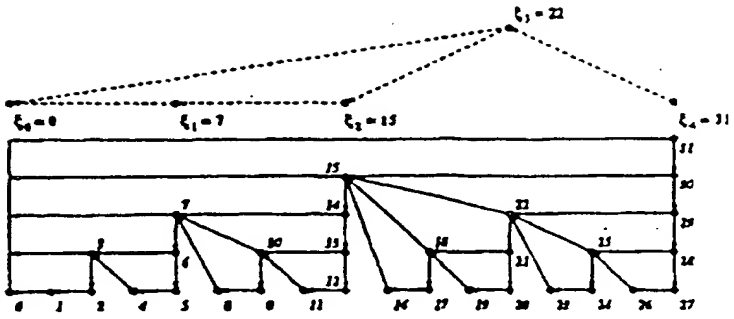

FIG 4

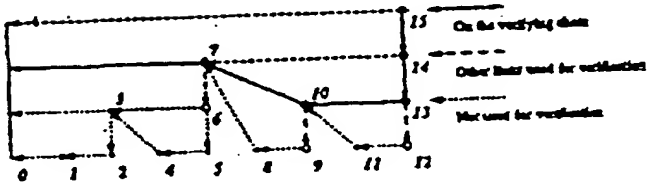

FIG 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/19061

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) : H04K 1/00; H04L 9/00
US CL :713/178, 168, 176
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 713/178, 168, 176; 380/23, 25

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS; (NONCE OR RANDOM NUMBER) (P) (TIMESTAMP? OR TIME (A) STAMP?) (P) (DIGIT? (2A) SIGN?)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,136,647 A (HABER et al.) 04 August 1992, THE WHOLE DOCUMENT.	1-17 --- 18-20
Y	US 5,422,953 A (FISCHER) 06 June 1995, THE WHOLE DOCUMENT	18-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search
24 OCTOBER 1999

Date of mailing of the international search report
04 NOV 1999

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
HRAYR A. SAYADIAN James R. Matthews
Telephone No. (703) 306-4169

Form PCT/ISA/210 (second sheet)(July 1992)*